WO 00/31644 4/PRTS 09/856813
JC Rec'd PCT/PTO 25 MAY 2001
PCT/AU99/01051

HIGH ASSURANCE DIGITAL SIGNATURES

Ins. a1

a1>

This invention relates to digital signatures and in particular to high assurance digital signatures.

sub a2> ~~Background~~

Signatures are used by people in all aspects of everyday life. As society moves inexorably into the age of computers and information technology, the need for a signature which can be used in the digital realm is rapidly becoming a necessity.

Digital signatures however are not new and have been described in the research literature for nearly 20 years, but are not yet capable of being described as "commonplace".

However, as more security critical information is exchanged digitally, and more importantly the economic value of the information and transactions being handled digitally becomes more important, the need for a secure digital signature is likewise increasing in importance.

For the use of digital signatures to become readily accepted in high value or high-risk applications, it is necessary for them to be secure and there are a number of aspects to this security which are necessary precursors to their commonplace use in the future:

1. The cryptographic algorithms used to generate signatures have to be complex enough to be unbreakable;

2. The users of the system need to have confidence in the public key distribution infrastructure;

3. The storage of private keys needs to be secure; and

4. The " endpoint" at which encryption is performed and signatures are created or validated needs to be secure.

The literature describes numerous successful attacks against security measures taken to ensure the abovediscussed issues.

The invention described herein relates in particular to the fourth aspect of the security issues mentioned above, by providing a secure "endpoint" for encryption and the creation and validation of digital signatures.

Endpoint attacks

Endpoint attacks affect the act of encryption and the creation or validation of digital signature, consequently the possibility of Endpoint attacks lower the confidence of the recipient of a digital messages that the message has been properly signed. The recipient may be unsure as to whether the digital message is original or indeed originates from the purported sender of the message.

An endpoint attack is different to a "protocol" attack which typically occurs during the transit of the message. In an endpoint attack, the attacker typically alters software on a sender's computer, so that the altered software modifies one or more messages which are being sent and received or initiates the sending of messages without the knowledge of the sender. Whereas in a protocol attack, an attacker eavesdrops on communications between the respective computers of the

sender and receiver and can impersonate either of the participants, or modify missives, etc.

There exist encryption techniques which can reduce or eliminate protocol attacks. However, endpoint attacks in the critical area between the user and their own computer are not aided by the encryption approach.

High assurance security and Endpoint attacks

Technology for building secure systems has mostly been developed in the military intelligence communities. In high-risk situations, policies require high assurance systems. That is, the users or owners of systems have to be highly assured, or confident, that the software and hardware systems they use will perform correctly. The consequences of failure can be so significant, that it is justifiable to spend substantial amounts of time and effort to achieve this assurance.

Assurance of this type is aimed partly at countering the threat of endpoint attacks and such assurance can be gained in a number of ways.

The most rigorous and objective methods, used by military and intelligence organisations, are described in publications such as TCSEC, ITSEC, and CC which provide a variety of physical, procedural and hardware and software approaches which for this specialist environment can provide the necessary assurances.

However, the majority of computers and computer systems used in critical environments do not meet the high levels of assurance which might be required by policy.

History has shown that the procurement of suitably secure and assured general purpose computer systems is very expensive and impractical. They are typically obsolete before they are delivered.

A device which provides high assurance digital signatures used as a limited-functionality peripheral provides a means for achieving high assurance security functions without the disadvantages associated with the development and evaluation of a much more complex, general purpose computer system.

Digital signature semantics

Signatures in the "paper" world (in contrast to the purely "digital" world) are used to indicate that the person who signs the document has written or read and thereby agrees with the content of the document and in accordance with its content, is bound by the fact of the signature to abide by that content. The term signature can also include the mark of a legal entity which may be represented in the form of a company seal applied by an authorised officer of the legal entity and typically countersigned by that person. Documents requiring signatures include, but are not restricted to, personal letters, contracts, or cheques.

Although there are many (surprisingly simple) ways to forge a signature, or undetectably modify the document which was signed, in critical circumstances there do exist well accepted procedures to increase the assurance that a signature has been applied by the appropriate person and that they were not under any duress at the time (eg witnesses).

Thus it is desirable that, when digital signatures are used for the purpose of positively identifying the person who applied the digital signature, the signed

digital message/document has the same legal value and effect as signatures used on a paper document.

As with paper documents, if a digital document is signed by a person or a legal entity, the recipient and reader of the document should be able to safely assume that the signer has written or at least read and agrees with the content of the document.

However, in a digital world it is even easier than in the paper world to change documents without detection, least likely but most importantly the creator of the (unsigned) document.

In the digital world attacks occur almost instantaneously and in a realm not physically examinable by its users, attacks can originate from those very same devices that users grow to rely on, "their own computer" and in a manner which can leave little, if any, no evidence of the mechanism or the attacker. Thus in the current digital world it is prudent not to place too much reliance on the veracity of a digitally signed message or document

Assumed threats

There are many computer systems today which can generate digital signatures for use with documents but they are all still vulnerable to "endpoint" attacks. The designers and users of digital signing programs are often unaware of this type of threat and for those that are aware, the confidence or trust that a recipient can place in the fact that a statement was signed is never high and potentially the whole system of digital signature and certificates will be placed into jeopardy if this threat is not properly addressed. This jeopardy increases as time passes by,

and becomes more critical as more and more users are seduced by the apparent surety of the current system.

One manifestation of a real and active endpoint threat is exemplified by the "Trojan horse" type attack. A Trojan horse is malicious software, of which a user of the computer is typically unaware or of when and how it operates. The Trojan horse software, as the name suggests, gains access to the memory of the computer being used, by (surreptitiously) accompanying the loading of a legitimate software program and once ensconced inside the computer, performs malicious functions, without the knowledge of the user.

It is advantageous to provide some explanation of the terms which will be often used in the description of the technology surrounding and defining the invention.

Public Key Cryptosystems (also known as Asymmetric Cryptosystems)
This is a well known art, described in many references.

In standard (symmetric, or secret key cryptosystems), the encryption and decryption operations use the same key. However, in public key cryptosystems, one key is used for encryption, and the decryption can only be performed with a certain other key. The two keys are related, and sometimes called a key pair. One key is usually designated a "public" key and the other a "private" key. The security of the system relies on the fact that it is computationally infeasible to derive that private key given the public key.

It is generally assumed that any participant in information exchanged can acquire a user's public key, but that the user will protect the private key to the best of their ability.

If the public key is used to encrypt some information, only the holder of the private key will be able to decrypt the information. This is useful for encrypting messages destined for a particular user, which should not be disclosed to any other user. If the private key is used to encrypt some information, any user will be able to decrypt the information with the public key. This will assure other users that it was only the holder of the private key who actually encrypted the message originally. This is useful for implementing a digital signature which comprises the transformation of the message using the private key to produce a string of digital data which uniquely represents the message and from which it is infeasible to decrypt the original message. Only the other of the key pair, in this case the corresponding public key, can determine whether the original document was used to create the signature. Furthermore, it is also possible to encrypt a message with a public key which can only be decrypted with the corresponding private key.

The use by a user of their private key is seen by most as the equivalent of physically signing the content of the message, as it is typically assumed that only the holder of the private key could encrypt it and only the public key could decrypt it. This also of course assumes that the public key verifiably corresponds to the private key.

For efficiency purposes, encryption of messages is frequently performed by using a symmetric algorithm (which is faster) to encrypt the message using a randomly generated message encryption key, then using the asymmetric algorithm to

encrypt the "message encryption key" with the recipient's public/private key. Only the recipient, with a corresponding public/private key, will be able to decrypt the "message encryption key", with which they can then decrypt the actual message.

Similarly, for signing a document, it is usual to calculate a "message digest" using a hash function. It is the digest (which is much shorter than the message) that is then encrypted with the signer's private key. This encrypted digest is called the signature. A recipient can decrypt the signature to recover the original digest value. The signature is valid if the hash of the message is equal to the decrypted signature.

Cryptographic Engine

A cryptographic engine is an electronic component which is able to perform the complex arithmetical and logical manipulations involving data and keys, to implement encryption, decryption, signature generation, and signature validation. A cryptographic engine may include a number of registers for storage of keys and/or intermediate results.

The designer of a cryptographic engine would typically expect that the engine would be used to perform various calculations in order to give senders and recipients various assurances about the confidentiality, integrity, or origin (or similar) of a message. This assurance can only be given if the engine operates correctly, even in the presence of certain threats (which the designer will typically be aware of). The engine should be trusted, for example, not to release unencrypted data when encrypted data is required.

Examples of cryptographic engines are the Capstone chip, a Fortezza Card, an encryption daughter board for a PC, and a software module in a program such as PGP.

Endpoint Attack

A typical endpoint attach may occur in the following manner.

Consider a user who wants to sign an e-mail message. The user's private key is typically stored in an encrypted state in a file on the hard disc of the user's personal computer. To create the signature, the user must enter a password or phrase which allows the file to be decrypted and the unencrypted private key temporarily stored in the computer so that the e-mail program can then perform the cryptographic calculations on the message to produce the signature using the private key.

The signature created is an upwards of 64 ASCII character length string which uniquely correlates the user's private key with the digital representations of the message.

Consider an endpoint attack which modifies the behaviour of the users e-mail program. It would be possible for the malicious program to read and then store the various keys used by the user (in particular the key strokes that comprise the pass word or phrase) in another location on the hard disc. Having knowledge of the pass phrase or a copy of the key/s allows the malicious program to sign other messages, which the user did not intend to sign. For example, messages authorising the purchase and shipping of goods to a unknown recipient could be created and signed, all without the authority or knowledge of the user. The Trojan horse program may also secretly communicate the private key or keys of

the user to another user, who could then fraudulently forge the user's digital signature onto any document supposedly sent by the original user.

This is the primary threat countered by the present invention.

This invention provides a means for securing against the unauthorised use of a user's private key which as a consequence provides high assurance that the digital signatures created by that key are legitimate. The invention can be embodied in a number of forms each of which is useful in different applications.

Sub a3

~~BRIEF DESCRIPTION OF THE INVENTION:~~

In a broad form of the invention a private key protection device (PKPD), comprises a digital private key storage means containing a user's digital private key; a cryptographic engine; a communications port for receiving digital data from an external device, and for transmitting data to said external device; a display means for displaying said received digital data; a user operable input means connected to said cryptographic engine to indicate when operated by said user their approval of said displayed received digital data; wherein said cryptographical engine is trusted to only apply said user's digital private key to sign received data only if said user operable input means is operated and communicate said signed data external of said digital private key protection device.

Specific embodiments of the invention will now be described in some further detail with reference to and as illustrated in the accompanying figures. These embodiments are illustrative, and not meant to be restrictive of the scope of the invention. Suggestions and descriptions of other embodiments may be included but they may not be illustrated in the accompanying figures or alternatively

features of the invention may be shown in the figures but not described in the specification.

BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 depicts a pictorial representation of elements of an embodiment of a private key protection device;

Fig. 2 depicts a further pictorial representation of elements of a further embodiment of a private key protection device;

Fig. 3 depicts the use of a message envelope created by the PKPD;

Fig. 4 depicts the use of a message envelope created by the PKPD;

Fig. 5 depicts the PKPD attached to a PC wherein the device has an in built display;

Fig. 6 depicts the PKPD attached to a PC wherein the device incorporates a keyboard, video and mouse pointer switching function; and

Fig. 7 depicts a network comprising devices attached to PC's in the network.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

When a user wishes to sign a document they created themselves or one that was received and has been amended, they use their private key which is typically stored on the hard drive of their personal computer PC. Their private key is in encrypted form so that not just anyone can locate it and use or copy it, so the

users will need to input a pass phrase known only to them to "unlock" use of the private key for the signing process.

As discussed previously, it is during the time the private key is "unlocked" that the PC and user are most vulnerable to an "endpoint" attack. Even if the key is contained in a supposedly more secure device, such as a smartcard or a PCMCIA card such as a Fortezza card, it will be apparent to the reader that malicious software will still be able to utilise the unprotected key or the cryptographic services of the device. As soon as the device is unlocked (by the user entering a PIN), the device will perform whatever calculations it is asked to perform, whether the user is aware of these or not.

The invention comprises a private key protection device (PKPD)which is capable of resisting "endpoint attacks" thus ensuring the safety of the private key at all times.

In its simplest embodiment the PKPD exists separate from the user's PC but connects to it and its peripherals as required via normal communication ports. The embodiment depicted in Fig. 1 shows the PKPD 10 receiving 12 and transmitting data 14 via a communications port 16 of a device 18 which could be a PC. In this embodiment the PKPD lies between the user 20 and the PC 18.

The PC can send any data to the PKPD, which may be for example a shopping order list, a military command to fire a missile, or a contract. The PKPD receives the data at its communications port 22 and directs the data to the display 24.

The display 24 displays to the user the data to be reviewed plus any other command or prompts required to enable the user to control the use of the "private key" which is stored in the private key storage means 26.

Once the user has read and understands the document displayed, if they do not agree or are not willing to authorise the meaning of its contents they can ignore it and the document will be removed by a reset of the PKPD or some such other approach which ensures that the document has been permanently removed from the PKPD.

If the user agrees with or authorises the contents of the document, just as they would in the paper domain, they can choose to digitally sign the document.

This is achieved using the PKPD by manually operating the user operable input means 28 which may be integrated with the PKPD. This input means may be labelled variously, an ACCEPT button, a SIGN button, etc and may have various mechanical properties. For example, it may be a momentary contact switch, it may be locked and unlocked for manual operation and inoperable without a physical key, it may be backlit when operable, it may require a predetermined depression sequence. These features are merely additional to the primary requirement of being user operable in the physical sense.

Once the user operable input means 28 is operated, the cryptographic engine 30 will access the user's private key in the private key storage means 26 and use it to generate a digital signature of the document and only that document which was displayed to the user.

Once the document is signed, it is transmitted to the PC 18 via the transmit data path 14 via both communications ports 22 and 16. A communications port is a device which transmits or receives digital data to/from another port using a common protocol. A communications port may comprise hardware (eg parallel "centronics" printer port RS232 protocol port) or a logical software arrangement (eg TCP/IP port). The transmission and reception of digital data could be conducted via wire or wireless means.

The electronic means to implement the above functions of the PKPD can be implemented by a person skilled in the art.

A PKPD counters endpoint attacks and assuming that the cryptographic algorithms are not breakable, and that there is confidence in the public key distribution infrastructure, it becomes possible to trust the use of private keys. It becomes accepted that the person purported to have applied the signature actually did apply it, and will therefore be bound by the content and intent of the digital document that was signed by that user. This type of assurance is sometimes referred to as "non-repudiation".

Thus the grocery store can be sure that the Jones' really do want the shopping list items in their message and if included, the authorisation of a credit card payment can be accepted without doubt prior to the shopping list items being delivered. The military order to fire a missile is real and if authorised by the appropriate military commander should be carried out without question.

Digital Certificate

A further use of a digital signature is to create a certificate which is a statement by the signer about another person or fact. A certificate is usually an indication

that the issuer of the certificate confirms certain information or that it has a particular property.

In the paper domain a certificate is something which attests to a fact, like graduation from University. The certificate is typically hard to forge as it will normally have a seal that can only be applied by or with the authority of the Chancellor of the University.

In practice there are other ways of verifying that a person has graduated from a University but the certificate is typically taken at face value.

Digital signatures can also be used to uniquely sign a document which acts like a certificate. In some ways a digital certificate is better than a paper certificate since a digital certificate can be checked by an equivalent device to that of a PKPD for decrypting and checking the veracity of the private key used to create the certificate as well as displaying the document which could not have been altered while bundled with the certificate.

The term certificate is frequently used to refer to a particular type of certificate, which is used to confirm a user's "public key" the second part of an asymmetric key system where the first part is the user's "private key". Such "public key" certificates are typically created by a third party, the third party being trusted by all users to take a user's (a user known to the required level by the third party) public key and bundle it into a "public key" certificate.

Thus the second user in possession of the first user's "public key" certificate can be sure it is actually that user's public key.

The reason for this procedure is to ensure that the second user does not use what they thought was the first user's public key when in fact it is the public key of a rogue user masquerading as the first user.

In such a masquerade it is not inconceivable for the rogue user to pretend to be the first user and dependent on the relationship (eg buyer/seller; general/soldier) the consequences of the second user being deceived could be catastrophic.

Thus, since the creation of a certificate can be very critical, it is best that the process is conducted on a PKPD which is immune to "endpoint" attacks.

Referring to Fig. 1 the various means displayed may not be provided in hardware but may be virtual means and implemented in software. For example, as stated previously, the communications port 22 may be a virtual TCP/IP port having a simple physical bus connection. A further example is the cryptographic engine which could be a function of a Central Processing Unit (CPU) which interacts with both on board and external memory and peripheral hardware.

However, it is preferable that the architecture of the PKPD be as simple a possible since the less hardware and software the more it is amendable to high assurance evaluation, as may be prescribed in the various documents mentioned previously to achieve an acceptable level of trust worthiness as relevantly defined.

The function of the PKPD which requires the greatest trust is that which ensures that the users key (private, public) or the PKPD's own key, is used once and only

once; used only to sign the data which has been displayed; and is only used if the user operable input means is activated by the user.

The way in which this functionality is created and the consequent high assurance evaluation of the PKPD are steps known to those skilled in the art after having been instructed of the arrangement of the invention.

The terms trust, assurance and confidence were used to describe desirable and preferable features of the PKPD and in the art these terms are generally considered to be synonymous but others may also be used.

Devices and systems including their methodology are often complex and interrelate with each other, humans and external influences in unexpected ways. However, device and system failure is unacceptable in so called critical situations and one such failure is in the failure of security in handling digital data. Designers of so called critical systems are obliged to demonstrate that their system has been implemented in such a way that the likelihood of failure is suitably low since it is very difficult to completely eliminate the likelihood of failure in any system.

Clearly in the digital domain, failure is not only by way of failure to perform a particular function it is also the designed ability to resist the persistent attack of hostile or mischievous system users who want to take advantage of weaknesses in the system.

The more critical the system is, the lower the acceptable likelihood of failure becomes and generally the lower it becomes the more expensive the system. Furthermore, the more complex the system becomes, the harder it is to achieve

an acceptably low likelihood of failure. Also, the more complex a system is the more difficult and expensive it is to evaluate the presence or absence of faults and bugs in the implementation.

Thus, we as humans, learn to allocate different levels of trust to various systems but the meaning of the term "trust" will always be a property of the context of its use and our understanding of the likelihood of failure.

Persons skilled in the art will generally recognise the property which makes a system trusted, but in some cases it will be necessary to explicitly define the property.

Thus, in a PKPD as stated previously, it is preferable that it be trusted to use a stored key on the displayed document and only used if the user operable input means is activated. Clearly the key must be secured from unauthorised access and may if desirable be stored in encrypted form.

In the case of a public key, it must be stored so that it can not be altered in an unauthorised way.

The display means 24 as depicted in Fig. 1 and used in other embodiments described in this specification is a device for converting data into a human readable form. Display technologies are everchanging examples of which include CRT monitors and LCD monitors. Other types of display include printers and even Braille output devices so that the sight impaired can perceive the data.

It is preferable in the context of the important use of a PKPD that the monitor be trusted to display exactly the data presented to it and that the displayed form of

the document be such that the user who observes the display can not, once having signed the data with a PKPD, declare that the data was not displayed accurately to them.

Figs. 5 and 6 depicts PKPD's 40 and 50 respectively, each having their own monitors 46 and 62 and further in Fig. 6 the PC's monitor 56 can be used to display exactly the data to be reviewed by the user of the PKPD.

Thus it may be that the data format is standardised in that all characters are a minimum size and predetermined font. It may also be important to filter the data to exclude all macro's or executable because it is important to be sure that only displayable data is reviewed and signed.

Importantly, it is highly preferable that only that data which is displayed is signed by the predetermined key held in safe storage in the key storage means.

The user operable input means 28 depicted functionally as a block in Fig. 1 could be in one of the various forms previously described, importantly, it will be apparent to those skilled in the art that the means itself is a typically mechanical one that requires interaction of the human user and thus not operable by a rogue program. This means that the signal generated by the switch is not capable of being replicated by any equivalent signal generated even within the PKPD itself. In a highly critical environment where a PC and keyboard can not be trusted. This clearly obviates the use of a software equivalent or any combination of keyboard based keys associated with the user's PC.

It is also clear that there is preferably a degree of physical security related to the PKPD and the times at which it is being used by the user. The PKPD may

preferably require the user to identify and authenticate to the PKPD before it functions as required. Such identification and authentication may involve the use of user id, password, pass phrase, PIN, token, biometric or other means, or combinations thereof.

In this regard, it is possible as will be described in other aspects of the invention that the key storage means is physically removable from the PKPD and although the keys are stored therein they can not be physically extracted, otherwise any attempt to do so will destroy the keys and possibly the removable memory device itself.

Furthermore, the operation of the PKPD can be predicated on the successful response to a challenge to the proper user by the PKPD. The challenge could be in the form of a question and a predetermined response by the user. The answer could require a further input to the PKPD in the form of a keyboard for example providing numeric or alpha numeric input for the response or a biometric response in the form for example of a iris check by an appropriate sensor device included in the PKPD device (not shown).

In addition to the physical requirement to operate the PKPD (eg insertion of a valid key storage means (eg SMART CARD or FORTEZZA CRYPTO CARD)) and a valid challenge response, it may also be preferable to operate an audit log of all transactions performed or attempted with the device.

An audit log comprises a collection, typically strictly chronological, of information representative of all transactions performed by the PKPD. Such a log will identify (typically after the fact) those transactions which should not have

taken place and thus unauthorised use of the signature by even the authorised user will be available for scrutiny.

Preferably security properties of an audit log include the inability to alter or clear the log and with this property intact it is possible to claim non-repudiation of the transactions in respect of the user performing the transaction.

Although, the PKPD has been described and likely understood to be useable by only one user, it is possible for a PKPD to be useable by multiple users as long as it can partition the separate user's keys or alternatively accept multiple insertable key storage means.

Fig. 2 depicts a further embodiment of the invention of a PKPD comprising similarly identified elements such as a communications port 22, a display 24, a user operable input means (accept key) 28 but elements such as a cryptographic engine (30 in Fig. 1) are replaced or enhanced by the use of a CPU 30a, a RAM 30b and a ROM 30C.

Additional elements comprise a smartcard interface 32 for the insertion of a smartcard containing a specific user's keys, and a video switch 34 for receiving a video signal from an attached PC, and passing that signal to the PC's monitor, except when the PKPD is active, in which case the PKPD's display information is passed to the monitor. Such an arrangement is depicted in Fig. 6 where the PKPD 50 is located physically between the user's PC 52 and its monitor 56, keyboard 54 and pointing device 48.

If the PKPD uses the PC's monitor for display of information to the user, it is preferable to have an additional unforgeable indicator which can be used to

inform the user whether the information displayed on the monitor is trusted (coming from the PKPD) or not (coming from the PC). An example of such an indicator may be a LED on the PKPD front panel, near the user operable input means 64 on Fig. 6.

There is also a Private Key Storage means 38 for the PKPD's own private key the use of which will be discussed later in the specification.

The video switch 34 of this embodiment is arranged to take over control of the user's own PC monitor and replicate the trusted display function described previously. Thus even if this embodiment did not have a display 24, it could still display the received data/document in a trusted manner on the user's PC monitor. Also refer to Fig. 6 for a depiction of such an arrangement.

The smart card reader 32 allows the PKPD to have a further level of security since the smart card containing the user's keys and other selected keys can be kept in a physically secure place until it is needed saving the need to physically secure the whole PKPD which invariably would have involved disconnecting and storing a more bulky device than a smart card if connected with cables (wireless PKPD's are also possible).

Furthermore as stated previously, multiple users can use the same PKPD.

Yet further this embodiment has the provisions to apply a signature unique to the PKPD itself. This provides further surety to the recipient that the user's signature was indeed created on a PKPD, and not on, say, an untrusted PC. This could indicate to the recipient that the originator was willing to be legally bound by the message, and that the originator would not be able to repudiate having

sent the message. (Contrast this with a message without the PKPD signature: the originator could establish reasonable doubt about the fact that he deliberately signed the information, by proposing the plausible scenario in which a virus or Trojan horse on his PC signs information with his signature, without his knowledge). The existence of a PKPD signature assures the recipient that a PKPD was used by the originator and furthermore increases the non repudiation factor of the originator.

The PKPD could be constructed in such a way as to interpret organisational policy before permitting information to be signed with an organisation's signature. For example, it may be that the PKPD will only sign a message with a company's private key (which is stored inside the PKPD) if at least two of the company's directors have individually signed the message.

A certificate issuer could use a PKPD to create certificates. The certificates could be signed twice, once using the issuer's personal private key, and the second time using the PKPD private key. Any person wishing to rely on such a certificate in the future could have greater confidence in the fact that the issuer deliberately intended the certificate to be signed, because of the presence of the PKPD signature. In contrast, if the issuer's PC was infiltrated by malicious software, that software could create any certificates it wanted, and if the issuer's private key were available to the PC (either directly, or via a smartcard or Fortezza card), the certificate could be signed with that key without the issuer's knowledge or consent.

Fig. 3 depicts a message text (document) which has been first signed by User B's public key (ie can only be decrypted (unbundled) by User B's private key) and the signed document is then signed again by the User B's PKPD public key (ie

the result can only be decrypted (unbundled) by a corresponding PKPD private key).

This arrangement provides a message which can only be decrypted with both the private key of User B and the private key of User B's PKPD. It would be possible to construct the PKPD to display the decrypted version of such messages to User B, but to ensure that the decrypted version was never released outside the PKPD. This would mean that although User B could view the decrypted information on his screen, he wouldn't be able to print it, save it on disk, or forward it to any other user. Such a property can be referred to as an "Eyes Only" property.

An improvement in this mechanism would be to use a PKPD shared key, instead of the User B PKPD's public key to perform the second encryption. This would allow a mobile User B to read the message at any convenient PKPD, rather than at only the specific one whose public key was used.

Fig. 4 depicts the same User B public key use so that the inner layer can only be read by User B but the outer layer is signed by the PKPD's shared key so that any PKPD can remove the outer layer.

In this example a Shared Key is used, meaning that the encryption which forms part of the signature process uses a single symmetric key held by every PKPD and thus only PKPD's having that shared key may remove the outer layer.

Outer and inner layers are akin to the inner and outer envelopes of the SAFE HANDS document protocol in the paper domain.

In a further embodiment there exists a mechanism for receiving the document/message from a computer, and transmitting the signed document to its next destination directly from the PKPD via its communications port using a physical layer transport mechanism such as Ethernet, serial, parallel, PCI, SBUS, SCSI, VSB etc. Thus this mechanism excludes the transmitted signed document travelling back to the computer from which it was received.

In a yet further embodiment of the PKPD it would not only perform a signature generation function but also be capable of validating signed documents received.

The process of validating a digital signature is subject to endpoint attack. For example, a malicious e-mail program could inform the user that the signature attached to a message was valid, when this was not the case. In fact, a malicious e-mail program could display to the user any message it wanted to, as well as asserting that the message was signed in a most trustworthy way, even though no such message had ever been signed or sent.

A PKPD can be used to counter such endpoint attacks. The process of validating a digital signature involves the use of a public key. The PKPD, on being presented with a signed message, can calculate the validity of the message with respect to the public key, and then display the message contents and the signature validity to the user.

In general, a PKPD would not be configured with a public key for every potential originator. Instead, a certificate hierarchy is likely to be used, involving a single "root" certificate (typically self-signed), from which other certificates can be validated, eventually assuring the relationship between the originator and the originator's public key. The PKPD, on being presented with a signed message for

validation, as well as the appropriate chain of certificates (typically retrieved from a directory) can calculate the validity of the certificate chain, as well as the validity of the message. The contents of the message and signature validity, as well as the certificate chain details, can be displayed to the user.

The protection of the root certificate is important. An attacker who could modify the root certificate could supply a complete chain of "bogus" certificates, which would be valid with respect to the modified root certificate. Current e-mail programs are vulnerable to this threat. A PKPD would have to store the root public key in a manner which would prevent its unauthorised modification. Physical and procedural means could be used to provide this security.

A yet further embodiment of the use of a PKPD is to incorporate into the message being signed an "indicator" which can act as a flag to network security devices that there is an authentically signed message within the outer layer/s. The network security devices then may allow the signed document to leave, let us say a high security network for a network of lower security. This "indicator" can indicate that the message has been sealed and that it is safe for the message to be transported via insecure communication means (eg the Internet) to its intended destination. It is thus possible to implement a multi-level secure (MLS) messaging system. Fig. 7 is an illustration of various PC's and associated PKPD's communicating via local and Internet networks.

In another embodiment the PKPD would be programmed to produce signatures using standard protocols, such as MSP,CSP (ACP-120),S/MIME,PGP, etc.. This would have the advantage that commercial off the shelf infrastructure used elsewhere would "understand" the signature, although it may not fully appreciate the high assurance nature of such signatures.

In some protocols, such as MSP, CSP, S/MIME, where there can be two signatures, the device can offer advantages over those mentioned above. The PKPD can create one signature using the user's private keys, which may be used for purposes unrelated to the devices' invention. A second signature can be created using special keys devoted solely to the function of the PKPD.

A user may have private keys which are used for a variety of functions apart from the ones described in this document. For example, keys may be used for the authentication and establishment of a remote login session over the Internet. Another embodiment of the PKPD could accept requests from the connected PC that would ordinarily have been dealt with by a smartcard or Fortezza card connected directly to that PC. The PKPD would preferably alert the user to the fact that the keys were being used (although it would not necessarily be able to indicate to the user for exactly what purpose they were being used), before performing the appropriate signature, validation, encryption, or decryption (or other) operation. Although such a system would potentially allow the user's key to be abused by malicious software, the PKPD's own private key would not be made available in the same way. Its use would be reserved for instances in which the user is able to make an informed, deliberate, and legally binding choice to sign the document after it has been displayed by the PKPD.

A PKPD will be able to create a second signature, using special keys, to indicate the high assurance nature of the signature. Preferably, the user's keys would be used to create an inner signature, which would be encompassed by the second signature created by, for example the Fortezza Crypto card, using the PKPD's private key.

A signature validating device being a slight variant of a PKPD can thus translate the twice signed document into a conventional document while providing the necessary assurance of its originator and the particular PKPD which applied the signatures.

In large financial institutions, it is not practical to have "manually" signed cheques. Instead, the signatures are printed (by machine) onto the cheques. If it were possible for malicious software to be introduced into such a system, cheques could be obtained improperly. It would be possible to use PKPDs to secure the cheque printing process. An authorised user would have to review the appropriate details for the cheques, and then "accept" them on the PKPD, which would sign a message containing the details. Another PKPD, connected directly to the cheque printer, would verify that every order to print a cheque had been signed by a PKPD. Since the graphic containing the authorising (printed) signature is stored only within this latter PKPD, it would not be possible to print illegitimate cheques without using the PKPD. This would counter the threat of malicious software. The printer will need to be connected directly to the PKPD. Since the graphic containing the authorising (printed) signature is stored only within the PKPD, cheques appearing with the particular graphic could only have been produced with a high assurance PKPD. Such a signature would be unique to the printed content of the cheque (be that the amount, the words describing the amount, the date, the time, the payee and a unique transaction number).

The PKPD could be programmed to display information in particular formats which are in a convenient human readable form. For example, if messages are written in Hyper Text Markup Language (HTML), the device could render the HTML, instead of showing all of the tags of the generated signature within the message.

High value electronic transactions may be required to be authorised using a high assurance PKPD and to facilitate easier transactions, the PKPD monitor would display Secure Electronic Transaction (SET) messages in a form convenient to the PKPD user.

The device of the invention is preferably able to check the authority of the user to sign certain types (eg. classifications) of messages, before proceeding to allow a user to do so. The user's authority could be stored with the keys, or in certificates communicated to the device with the message or at any other time.

The device of the invention is preferably able to encrypt information being transmitted, in order to preserve the confidentiality of the message content until it is decrypted by the recipient.

High assurance digital signature device

A preferred embodiment of the high assurance digital signature device comprises an embedded microprocessor which executes a program stored in ROM. Preferably the ROM and microprocessor are mounted on the same integrated circuit chip and arranged so that elements within the circuit cannot be changed so that the integrity of the device as created can not be interfered with.

Interfaces

In a preferred embodiment there are three operational interfaces:

Network Interface

In one embodiment the device contains an Ethernet network interface, and communications with the user's personal computer occurs over the Ethernet.

The choice of this network protocol allows the device to be used with a wide variety of personal computers, work-stations, X-terminals, and other networked computers.

In some environments, it may be possible to assume that all user's computers will have, for example, SCSI or bi-directional parallel interfaces. There is also no reason that these could not be used for communication with the device.

User Interface

In a further embodiment interfaces to the user may be categorised as **bulk** inputs or outputs or **Boolean** inputs or outputs.

1. A Boolean output interface could be as simple as a light emitting diode (l.e.d.) Such an output only needs to indicate whether the bulk output interface is active or not. When the l.e.d. is lit, it may indicate that the device has taken over the user's screen as described in a previous embodiment.

2. The preferred bulk input interface is the keyboard of the user's computer. In other trusted systems a keyboard switch has been used to divert the output data from the keyboard to a different destination. Similar technology is used in this embodiment to divert the representations of keystrokes on the keyboard or other input device to the PKPD, instead of the user's computer. An alternative mechanism is to have a keypad or keyboard built into the PKPD. A bulk input device is used for the entry of data, such as, personal identification (PIN) phrases etc.

3. The preferred bulk output interface is the monitor of the user's personal computer. Just as the keyboard switch described above is used to "takeover" the

keyboard, a video switch is preferably used to allow the device to "takeover" the monitor. Any output displayed on the monitor by the device can be trusted to be the output provided from the bulk input interface. The user can tell whether the information on the monitor is that supplied from the device or not, by checking the status of the Boolean output which, for example, may be a light emitting diode (l.e.d.). A monitor may also be built into the PKPD. Figs 5 and 6 display these two arrangements. The PKPD 40 in Fig 5 is merely connected to the users PC 42 communication port and has its own screen 46 and a user input means 44.

4. A preferred Boolean input device is a simple push button switch mounted on the device. With such a switch, the user can provide a positive indication to the device that the document being displayed on the bulk output interface is acceptable. This switch is referred to previously as a user operable input means.

User's Fortezza Crypto card Interface

In one embodiment the user is able to insert their Fortezza Crypto card into the device. This allows the PKPD to authenticate the user, by checking that the user has entered the correct PIN phrase for the Fortezza Crypto card, and it also provides a convenient secure storage for the user's private keys used for encrypting messages if need be.

Functions of the PKPD

In one embodiment the PKPD can be arranged to provide a limited range of functions. Using the "client -- server" model, the PKPD can be described as a server. Note that this does not imply that it is a large machine, located in a special room, and shared by many users at once. In this embodiment it means that the PKPD does not initiate actions itself, it only responds to requests from

another system or device, which is for the purposes at hand referred to as a client.

The client, typically a user's personal computer, sends requests to the PKPD. After performing the appropriate function as determined by the request, the PKPD sends a reply back to the client.

In a preferred embodiment a number of functions which could be offered by such a device include login; set personality; submission and delivery.

An important aspect of the submit function is the secure manner of reviewing and signing operations conducted by the PKPD. The PKPD is arranged to make it impossible for the message to be modified between the reviewing and signing steps of the process but this does not imply that after reviewing the message the signing function must occur.

Secure Messaging

The following is a description of an embodiment of the way in which a high assurance digital signature device and method of use can be integrated into an existing messaging system which uses the MSP (Message Security Protocol) and a Fortezza Crypto card. A similar approach could be used for systems using other protocols.

Existing system

The typical messaging system incorporating MSP performs the login and message submission processes as follows which is in accordance with MSP ICD. Note that although the order of some steps is important, the precise order described herein is not the only valid process.

1. The user starts a Messaging User Agent (MUA) program, such as Netscape, Exchange, or Notes. As part of the start-up sequence; the user is required to login to the Fortezza Crypto card. The MUA program provides a pop up box, into which the user is asked to enter a PIN phrase. The PIN phrase is passed as an argument to the MSP_ LOGIN call.

2. The MSP_LOGIN function passes the PIN Phrase to the Fortezza Crypto card, which verifies it, thus authenticating the user.

3. The MSP_LOGIN function instructs the Fortezza to provide a list of personalities, whose private keys are stored on the card.

4. The MSP_LOGIN function returns, passing the list of personalities as a result.

5. The MUA program displays the list of personalities to the user in another dialog box. The user is invited to select one of the personalities with whose key, messages will be signed, encrypted, or decryption.

6. The MUA program passes the selected personality as an argument to the MSP_SETPERSONALITY function.

7. The MSP_SETPERSONALITY function instructs the Fortezza Crypto card to select the appropriate private key.

8. When the user chooses to send a new message the MUA program creates a window for the new message, and the user chooses a recipient/s (either by

typing in addresses or choosing them from an address book), types in the message, and adds attachments if necessary. The user then selects the required security services: either none, sign, or sign and encrypt. When the composition is complete, the "Send" button is activated. The user's computer may or may not have control of the user's ability to attach certain files.

9. Submission access control then occurs.

10. The MUA program then begins the process of invoking none or "one or more" security options. For example, the Fortezza Crypto card is instructed by the MUA program to calculate a hash value (eg. MD5) and signature, and if appropriate, to encrypt the message. The signature and plain or encrypted message are constructed into a "Protocol Data Unit" (PDU) according to the protocol.

11. The PDU is attached to header or "envelope" information, which is then transferred to the messaging server, or Message Transfer Agent (MTA).

12. The typical delivery and verification/decrypted mechanism then follows.

Modifications to provide a high assurance mechanism

The following steps show the modifications required to change the existing system so that it can support the high assurance mechanism.

1. When the MUA is loaded, instead of loading in the standard MSP software as an integral part, software is loaded to allow the MUA to communicate with the PKPD. When the MUA calls the "MSP _ LOGIN" function, the software sends a signal to the device to indicate that the login

function should commence. The device allows the user to enter the PIN phrase through a built-in keypad (or through the computer's keyboard, if keyboard switching functionality is included).

2. The device returns a signal to the computer, including the listed personalities. The software receives the signal, and "returns" the list as the result of the MSP _ LOGIN call. When the user chooses a personality, the MUA is loaded.

3. When the MUA software would normally call the MSP_SETPERSONALITY function, the appropriate parameters are instead transmitted to the PKPD. The PKPD displays the chosen personality to the user, and allows the user to verify that this is appropriate. This protects the user against the threat of malicious software choosing an inappropriate personality for messages about to be signed by the user. The "results" of the MSP_SETPERSONALITY function would be returned to the MUA.

4. When the MUA would previously have requested the card to hash and sign particular data, the function would instead be routed to the PKPD, in a similar way. The information being signed would be displayed to the user, and the user would have to indicate acceptance before the signature value was released back to the MUA. To perform this function, the PKPD may have to interpret appropriate protocol data units in order to present a human readable version of the message to the user.

A preferable high assurance digital signature mechanism provides one or more of the following features:

1. A message which is signed with a high assurance signature should not be vulnerable to Trojan horse software attacks on the computers of either the originator or receiver. That is, it is assumed that the software on all the systems have been maliciously changed to subvert the security of the computer but even those malicious changes will not affect the functions of a PKPD in its signing or verification functions.

2. Subject to the strength of the cryptographic algorithms it should be impossible to forge a high assurance signature.

3. The high assurance signature can preferably be conveyed within standard protocols, thus allowing a user with Commercial Off the Shelf (COTS) standard compliant software to receive messages from, and transmit messages to, a user using a High Assurance Digital Signature device and method.

4. The user should preferably be allowed to use their Fortezza Crypto card in an un-trusted computer for un-trusted applications.

It will be appreciated by those skilled in the art, that the invention is not restricted in its use to the particular application described and neither is the present invention restricted in its preferred embodiment with regard to the particular elements and/or features described or depicted herein. It will be appreciated that various modifications can be made without departing from the principles of the invention, therefore, the invention should be understood to include all such modifications within its scope.